

UNITED STATES PATENT APPLICATION

FOR

**IMPROVED SUBPIXEL RENDERING FILTERS FOR HIGH BRIGHTNESS SUBPIXEL
LAYOUTS**

BY

Candice Hellen BROWN ELLIOTT

AND

Michael Francis HIGGINS

**IMPROVED SUBPIXEL RENDERING FILTERS FOR HIGH BRIGHTNESS SUBPIXEL
LAYOUTS**

BACKGROUND

[01] In commonly owned United States Patent Applications: (1) United States Patent Application Serial No. 09/916,232 (“the ‘232 application”), entitled “ARRANGEMENT OF COLOR PIXELS FOR FULL COLOR IMAGING DEVICES WITH SIMPLIFIED ADDRESSING,” filed July 25, 2001; (2) United States Patent Application Serial No. 10/278,353 (“the ‘353 application”), entitled “IMPROVEMENTS TO COLOR FLAT PANEL DISPLAY SUB-PIXEL ARRANGEMENTS AND LAYOUTS FOR SUB-PIXEL RENDERING WITH INCREASED MODULATION TRANSFER FUNCTION RESPONSE,” filed October 22, 2002; (3) United States Patent Application Serial No. 10/278,352 (“the ‘352 application”), entitled “IMPROVEMENTS TO COLOR FLAT PANEL DISPLAY SUB-PIXEL ARRANGEMENTS AND LAYOUTS FOR SUB-PIXEL RENDERING WITH SPLIT BLUE SUB-PIXELS,” filed October 22, 2002; (4) United States Patent Application Serial No. 10/243,094 (“the ‘094 application), entitled “IMPROVED FOUR COLOR ARRANGEMENTS AND EMITTERS FOR SUB-PIXEL RENDERING,” filed September 13, 2002; (5) United States Patent Application Serial No. 10/278,328 (“the ‘328 application”), entitled “IMPROVEMENTS TO COLOR FLAT PANEL DISPLAY SUB-PIXEL ARRANGEMENTS AND LAYOUTS WITH REDUCED BLUE LUMINANCE WELL VISIBILITY,” filed October 22, 2002; (6) United States Patent Application Serial No. 10/278,393 (“the ‘393 application”), entitled “COLOR DISPLAY HAVING HORIZONTAL SUB-PIXEL ARRANGEMENTS AND LAYOUTS,” filed October 22, 2002; (7) United States Patent Application Serial No. 01/347,001 (“the ‘001 application”)

entitled "IMPROVED SUB-PIXEL ARRANGEMENTS FOR STRIPED DISPLAYS AND METHODS AND SYSTEMS FOR SUB-PIXEL RENDERING SAME," filed January 16, 2003, each of which is herein incorporated by reference in its entirety, novel sub-pixel arrangements are disclosed for improving the cost/performance curves for image display devices.

[02] For certain subpixel repeating groups having an even number of subpixels in a horizontal direction, the following systems and techniques to affect improvements, e.g. proper dot inversion schemes and other improvements, are disclosed and are herein incorporated by reference in their entirety: (1) United States Patent Application Serial Number 10/456,839 entitled "IMAGE DEGRADATION CORRECTION IN NOVEL LIQUID CRYSTAL DISPLAYS"; (2) United States Patent Application Serial No. 10/455,925 entitled "DISPLAY PANEL HAVING CROSSOVER CONNECTIONS EFFECTING DOT INVERSION"; (3) United States Patent Application Serial No. 10/455,931 entitled "SYSTEM AND METHOD OF PERFORMING DOT INVERSION WITH STANDARD DRIVERS AND BACKPLANE ON NOVEL DISPLAY PANEL LAYOUTS"; (4) United States Patent Application Serial No. 10/455,927 entitled "SYSTEM AND METHOD FOR COMPENSATING FOR VISUAL EFFECTS UPON PANELS HAVING FIXED PATTERN NOISE WITH REDUCED QUANTIZATION ERROR"; (5) United States Patent Application Serial No. 10/456,806 entitled "DOT INVERSION ON NOVEL DISPLAY PANEL LAYOUTS WITH EXTRA DRIVERS"; (6) United States Patent Application Serial No. 10/456,838 entitled "LIQUID CRYSTAL DISPLAY BACKPLANE LAYOUTS AND ADDRESSING FOR NON-STANDARD SUBPIXEL ARRANGEMENTS"; (7) United States Patent Application Serial No. 10/696,236 entitled "IMAGE DEGRADATION CORRECTION IN NOVEL LIQUID CRYSTAL

DISPLAYS WITH SPLIT BLUE SUBPIXELS”, filed October 28, 2003; and (8) United States Patent Application Serial No. 10/807,604 entitled “IMPROVED TRANSISTOR BACKPLANES FOR LIQUID CRYSTAL DISPLAYS COMPRISING DIFFERENT SIZED SUBPIXELS”, filed March 23, 2004.

[03] These improvements are particularly pronounced when coupled with sub-pixel rendering (SPR) systems and methods further disclosed in those applications and in commonly owned United States Patent Applications: (1) United States Patent Application Serial No. 10/051,612 (“the ‘612 application”), entitled “CONVERSION OF RGB PIXEL FORMAT DATA TO PENTILE MATRIX SUB-PIXEL DATA FORMAT,” filed January 16, 2002; (2) United States Patent Application Serial No. 10/150,355 (“the ‘355 application”), entitled “METHODS AND SYSTEMS FOR SUB-PIXEL RENDERING WITH GAMMA ADJUSTMENT,” filed May 17, 2002; (3) United States Patent Application Serial No. 10/215,843 (“the ‘843 application”), entitled “METHODS AND SYSTEMS FOR SUB-PIXEL RENDERING WITH ADAPTIVE FILTERING,” filed August 8, 2002; (4) United States Patent Application Serial No. 10/379,767 entitled “SYSTEMS AND METHODS FOR TEMPORAL SUB-PIXEL RENDERING OF IMAGE DATA” filed March 4, 2003; (5) United States Patent Application Serial No. 10/379,765 entitled “SYSTEMS AND METHODS FOR MOTION ADAPTIVE FILTERING,” filed March 4, 2003; (6) United States Patent Application Serial No. 10/379,766 entitled “SUB-PIXEL RENDERING SYSTEM AND METHOD FOR IMPROVED DISPLAY VIEWING ANGLES” filed March 4, 2003; (7) United States Patent Application Serial No. 10/409,413 entitled “IMAGE DATA SET WITH EMBEDDED PRE-SUBPIXEL

RENDERED IMAGE” filed April 7, 2003, which are hereby incorporated herein by reference in their entirety.

[04] Improvements in gamut conversion and mapping are disclosed in commonly owned and co-pending United States Patent Applications: (1) United States Patent Application Serial No. 10/691,200 entitled “HUE ANGLE CALCULATION SYSTEM AND METHODS”, filed October 21, 2003; (2) United States Patent Application Serial No. 10/691,377 entitled “METHOD AND APPARATUS FOR CONVERTING FROM SOURCE COLOR SPACE TO RGBW TARGET COLOR SPACE”, filed October 21, 2003; (3) United States Patent Application Serial No. 10/691,396 entitled “METHOD AND APPARATUS FOR CONVERTING FROM A SOURCE COLOR SPACE TO A TARGET COLOR SPACE”, filed October 21, 2003; and (4) United States Patent Application Serial No. 10/690,716 entitled “GAMUT CONVERSION SYSTEM AND METHODS” filed October 21, 2003 which are all hereby incorporated herein by reference in their entirety.

[05] Additional advantages have been described in (1) United States Patent Application Serial No. 10/696,235 entitled “DISPLAY SYSTEM HAVING IMPROVED MULTIPLE MODES FOR DISPLAYING IMAGE DATA FROM MULTIPLE INPUT SOURCE FORMATS”, filed October 28, 2003 and (2) United States Patent Application Serial No. 10/696,026 entitled “SYSTEM AND METHOD FOR PERFORMING IMAGE RECONSTRUCTION AND SUBPIXEL RENDERING TO EFFECT SCALING FOR MULTI-MODE DISPLAY” filed October 28, 2003.

[06] Additionally, these co-owned and co-pending applications are herein incorporated by reference in their entirety: (1) United States Patent Application Serial No. [ATTORNEY

DOCKET NUMBER 08831.0064] entitled “SYSTEM AND METHOD FOR IMPROVING SUB-PIXEL RENDERING OF IMAGE DATA IN NON-STRIPED DISPLAY SYSTEMS”; (2) United States Patent Application Serial No. [ATTORNEY DOCKET NUMBER 08831.0065] entitled “SYSTEMS AND METHODS FOR SELECTING A WHITE POINT FOR IMAGE DISPLAYS”; (3) United States Patent Application Serial No. [ATTORNEY DOCKET NUMBER 08831.0066] entitled “NOVEL SUBPIXEL LAYOUTS AND ARRANGEMENTS FOR HIGH BRIGHTNESS DISPLAYS”; (4) United States Patent Application Serial No. [ATTORNEY DOCKET NUMBER 08831.0067] entitled “SYSTEMS AND METHODS FOR IMPROVED GAMUT MAPPING FROM ONE IMAGE DATA SET TO ANOTHER”; which are all hereby incorporated by reference. All patent applications mentioned in this specification are hereby incorporated by reference in their entirety.

BRIEF DESCRIPTION OF THE DRAWINGS

[07] The accompanying drawings, which are incorporated in, and constitute a part of this specification illustrate exemplary implementations and embodiments of the invention and, together with the description, serve to explain principles of the invention.

[08] **FIGS. 1 through 3B** are embodiments of high brightness layouts for displays of all types as made in accordance with the principles of the present invention.

[09] **FIG. 4** is one exemplary embodiment of a resampling of one of the color planes for one of the above high brightness layouts.

[010] **FIGS. 5A and 5B** are yet other embodiments of a high brightness layout for displays as made in accordance with the principles of the present invention.

[011] **FIG. 6** is one exemplary embodiment of a resampling of one of the color planes for the layout as shown in **FIG. 5**.

[012] **FIGS. 7 and 8** are yet other embodiments of high brightness layouts for displays as made in accordance with the principles of the present invention.

[013] **FIG. 9** is one exemplary embodiment of a resampling of one of the color planes for the layout as shown in **FIG. 8**.

[014] **FIG. 10** is one example of a reconstruction grid being superimposed onto a target 3-color subpixel layout.

[015] **FIGS. 11 through 14C** are examples of various resample areas depending on the relative positioning of input image data grid to target subpixel layout.

[016] **FIG. 15** is another embodiment of the relative position of a 3-color target subpixel layout shifted with respect to an input image data grid.

[017] **FIG. 16A through 18C** are examples of various resample areas for the example of **FIG. 15**.

DETAILED DESCRIPTION

[018] Reference will now be made in detail to implementations and embodiments, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

Subpixel Rendering for Five Color systems w/white

[019] Figure 1 shows one embodiment of a portion of a high-brightness, multiprimary display 100 substantially comprising a subpixel repeating group 102 as shown. Group 102 is an octal subpixel repeating group comprising white (or no color filter) subpixels 104, red subpixels 106, green subpixel 108, blue subpixels 110 and cyan subpixels 112. The white subpixel is added to help achieve the high brightness performance of the display. Additionally, as the white subpixels are good candidates for being centers of luminance for subpixel rendering (SPR) – the white, as the majority subpixel, gives high MTF Limit performance. In this embodiment, there are equal numbers of red, green, cyan, and blue subpixels – of course, other embodiments may deviate some from this color partitioning. Given that the white subpixel is adding brightness to the system and that the use of the cyan color is to give a wider color gamut, it may be advantageous to set the color points of the minority subpixels to be deeply saturated to result in a wide color gamut. It should be noted that these color points and energies are only “substantially” the colors described as “red”, “green”, “blue”, “cyan”, and “white”. The exact color points may be adjusted to allow for a desired white point when all of the subpixels are at their brightest state.

[020] Figure 2 shows a portion of another embodiment of a high brightness, 5-color display. Here, the subpixel repeating group is group 202 – which is larger than the one shown in Figure 1 because the color subpixels are placed on a hexagonal grid. One possible advantage of a hexagonal grid is that it tends to scatter the Fourier energies in more directions and points. This may be especially useful for the dark luminance wells caused by the blue subpixels. Another possible advantage is that each row contains all four colors as well as the white

subpixels, allowing for horizontal lines to be black and white, fully sharpened, without chromatic aliasing.

[021] One possible embodiment of a display system using this layout may process image data and render it as follows:

- (1) Convert conventional data (e.g. RGB, sRGB, YCbCr, or the like) to RGBCW+L image data, if needed;
- (2) Subpixel render each individual color plane;
- (3) Use the “L” (or “Luminance”) plane to sharpen each color plane.

[022] The subpixel rendering filter kernels may be constructed from area resampling theory, as disclosed earlier in many incorporated applications noted above. Both layouts may be subpixel rendered from data sets that have a one-to-one mapping. That is to say, one incoming conventional pixel maps to one white subpixel. The white subpixels may then fully reconstruct the bulk of the non-saturated luminance signal of the image. The surrounding colored subpixels then operate to provide the color signal. The incoming image may be any format of color signal, as long as color gamut mapping with or without gamut expansion, may operate to convert said format to RGBCW expected by the subpixel rendering engine. It will be appreciated that such area resampling filters may be replaced by other suitable subpixel rendering techniques: resampling using bicubic filter, sinc filters, windowed-sinc filter and any convolutions thereof. It will be further appreciated that the scope of the present invention encompasses the use of these other techniques.

[023] As the white subpixels are mapped one to one, they may use a unity filter with no further processing required. The color planes may be filtered using several possible kernels.

For example, assuming that the image is band-limited, one embodiment might shift the phase of each of the color planes and the Luminance plane to the interstitial positions of the color subpixels in the horizontal direction. This may be accomplished with a simple cubic interpolation filter: $-\frac{1}{16}, \frac{9}{16}, \frac{9}{16}, -\frac{1}{16}$. It should be noted that the white plane may not need to be shifted. For non-band-limited images (e.g. text or sharp edges in images), there may not need to be the cubic filtered phase shift as above.

[024] Then, the color planes may be filtered with an area resample filter. A Difference of Gaussian (DOG) filter applied to luminance may optionally be added, examples are given here:

1	2	1	
2	4	2	
1	2	1	(Divide by 16)

Area Resample Filter for hexagonal and square arrangement

0	0	-2	0	0
-1	0	0	0	-1
0	0	8	0	0
-1	0	0	0	-1
0	0	-2	0	0

(Divide by 16)

DOG Filter for hexagonal arrangement of FIG. 2.

[025] It should be noted that non-zero values coincide with the same color to keep the color balanced. Using the luminance signal implements a simplified “cross-color” sharpening.

[026] In another embodiment, one could also perform actual cross-color sharpening, distributing the values of the cross-color coefficients among the color filter kernels such that the matrices add up to the desired numbers such as above. One method that may be useful is to divide the values of the actual subpixel luminances -- red, green, blue, and cyan -- by the luminance value of the color that is being sharpened then multiply it by the matrix above times a suitable normalization constant such that it adds up to the matrix above. Another way might be to not perform the normalization, which would mean that some colors would experience greater than unity gain sharpening. The colors that experienced the greatest gain would be the colors with the lowest luminance. This last property may be useful to reduce the “dottiness” of the high spatial frequency detail, increasing the quality of the signal. These methods and techniques of using varying sharpening gain on the colors may also be driven by the luminance signal as above.

[027] In one embodiment, multiplying the values of the sharpening matrix by a constant allows adjustment of the gain of the system. For this embodiment, if the constant is less than one, the filter is softer; if the constant is greater than one, the filter is sharper. Of course, other embodiments are contemplated by the present invention with different matrices and constants.

[028] It should also be noted that the one possible method uses the simplest subpixel rendering filter kernels -- with the math being performed substantially by bit shift division and addition. Other methods and embodiments may give numbers that require more multi-bit

precision multipliers. Of course, performing the color gamut mapping may require such multipliers as well.

[029] As well as cross-color sharpening, one embodiment of the system may be implemented using self-sharpening by adding the two matrices together. For example, the following may be useful for the arrangement of FIG. 2:

0	0	-2	0	0	
-1	1	2	1	-1	
0	2	12	2	0	
-1	1	2	1	-1	
0	0	-2	0	0	Divide by 16

[030] Since the mapping of the conventional pixel data, in what ever form it comes in, to the multi-primary space is indeterminate, this may introduce a degree of freedom that could be advantageous. For example, choosing any given algorithm may always give the right color over all; but may not give the best visual result. For example, the color subpixels, not all having the same luminance, may introduce a spurious pattern for many non-optimal mappings. The desired color mapping would give the most even texture for patches of a given color, minimizing visible spatial frequencies of luminance modulation, over the broadest range of colors; hue, saturation, and brightness. Such a mapping would allow the fine details to be displayed using the algorithm disclosed above. In another embodiment, the system might work

with a plurality of transform matrices, if no single transform matrix provides optimal result for all colors. It may be advantageous to create domains, or even continuously variable transforms.

Rendering Novel RGBW Panels

[031] In many cases, novel RGBW panels (and 5-,6-, n- color panels, for that matter) will be called upon to render legacy RGB or other 3-color image data. In many applications incorporated by reference above, there are described various embodiments for subpixel rendering resampling a modified conventional image data set. The modification is that each and every incoming conventional pixel has four (or more) -- instead of three -- color component values; e.g. Red, Green, Blue, and "White". The "White" in quotes denotes that this color point may or may not be at the white point of the display when all color subpixels are set to their maximum values. It may be desirable that any Gamut Mapping Algorithm (GMA) conversion from RGB to RGBW (or other multiprimary color space) occur before the subpixel rendering to keep the image from being blurred. The filter set could be designed to produce good results for both text and photographs. For example, in the '094 application incorporated by reference, there is shown some novel RGBW and RGBC layouts. For these layouts, one embodiment of the filters for the SPR for layouts that have a red/green checkerboard such as shown in:

Red and Green use:

$$\begin{array}{ccccccc}
 -.0625 & 0 & -.0625 & & 0 & .125 & 0 & & -.0625 & .125 & -.0625 \\
 0 & .25 & 0 & + & .125 & .5 & .125 & = & .125 & .75 & .125 \\
 -.0625 & 0 & -.0625 & & 0 & .125 & 0 & & -.0625 & .125 & -.0625
 \end{array}$$

$$\text{DOG Wavelet} \quad + \quad \text{Area Resample} \quad = \quad \text{Cross-Color Sharpening Kernel}$$

[032] The Red and Green color planes are area resampled to remove any spatial frequencies that will cause chromatic aliasing. The DOG wavelet is used to sharpen the image using the cross-color component. That is to say, the red color plane is used to sharpen the green subpixel image and the green color plane is used to sharpen the red subpixel image. This allows the cross-color luminance signal to be impressed onto the color subpixels, ‘filling in the holes’ in color images. It should be noted that for monochromatic images, the results of cross-color DOG wavelet sharpening is the same as self-color sharpening. It should also be noted that the coefficients disclosed above are exemplary of one particular embodiment and that the present invention contemplates many other matrices having suitable coefficients that suffice.

[033] The Blue color plane may be resampled using one of a plurality of filters. For example, blue could be resampled with a simple 2 X 2 box filter:

.25 .25

.25 .25

[034] Alternatively, the Blue color plane could be resampled using a box-tent filter centered on the blue subpixel:

.125 .25 .125

.125 .25 .125

[035] Moreover, the white plane could also be filtered using one of a plurality of filters. For example, the white or cyan color plane could be resampled using a non-axis-separable 4 X 4 box-cubic filter:

	$-\frac{1}{32}$	$-\frac{1}{32}$	
$-\frac{1}{32}$	$\frac{10}{32}$	$\frac{10}{32}$	$-\frac{1}{32}$
$-\frac{1}{32}$	$\frac{10}{32}$	$\frac{10}{32}$	$-\frac{1}{32}$
	$-\frac{1}{32}$	$-\frac{1}{32}$	

[036] Alternatively, to help abate that there is no phase error, nor aliasing, on the white or cyan subpixel, an axis-separable 3 X 4 tent-cubic filter might be used:

$$\begin{array}{ccc}
 -1/64 & -1/32 & -1/64 \\
 9/64 & 9/32 & 9/64 \\
 9/64 & 9/32 & 9/64 \\
 -1/64 & -1/32 & -1/64
 \end{array}$$

[037] The use of the box-cubic and tent-cubic filters may help to reduce the moiré artifacts in photographs while maintaining sharpness in text by taking advantage of the mid-position of the white subpixels. Although not necessary, it is possible to use the same filters for both blue and white color planes. One could use either the plain box or tent for both, or the box-cubic or tent-cubic for both. Alternatively, the cubic filters should be chosen for both.

[038] Figures 3A and 3B show embodiments of a high brightness display having the repeating subpixel groupings as shown. Although these layouts may have any aspect ratio possible, Figures 3A and 3B depicts this layout with all subpixels having a 1:3 aspect ratio. That produces subpixels that are taller and thinner than a possible square outline or 2:3 aspect ratio. This layout comprises a combination where the blue sub-pixels have the same size as the red and green and the same number -- which results in a substantially color-balanced RGBW layout, since there is the same area coverage of the red, green, and blue emitters using the same filters as would be found in conventional RGB display panels.

[039] The layouts of Figures 3A and 3B have a potential advantage in that it may be manufactured on a standard RGB stripe backplane with a suitable change in the color filter.

One embodiment of a panel having one of these layouts may use any suitable form of SPR algorithm, as discussed herein or in applications incorporated by reference.

[040] In one embodiment, the image source data to the display might assume a square aspect ratio – thus, with no scaling, each input pixel would map to three sub-pixels in this layout. However, these RGBW 1:3 layouts are 4 sub-pixels wide per repeat cell. If source pixels are mapped to groups of three such sub-pixels, then three of the layouts tiled horizontally might suffice before all the possible combinations are found. For each different combination of three output sub-pixels grouped like this, a different set of area resample filters might suffice. This is similar to the process of finding a repeat cell and generating different sets of filters for scaling, as disclosed in applications incorporated above.

[041] In fact, the same logic that does scaling might be used to select suitable filters. In one embodiment, there could be a simplification that may be easier to implement than scaling. As in scaling, there may be symmetries that reduce the total number of filters, and in this case, there are only three filters that are used over and over again in different combinations of colors. Figure 4 depicts the resample areas and filters so generated for the red subpixels. The filters for green, blue and white are identical, but appear in a different order or orientation.

[042] As may be seen in Figure 4, the resample areas may be hexagons with three different alignments: offset 1/3 to the left (as seen as areas 404), centered (as seen as areas 406), or offset 1/3 to the right (as seen as area 402). The three resulting unique area resampling filters are:

2 12 0	0 14 0	0 2 12
82 146 0	22 184 22	0 146 82
2 12 0	0 14 0	0 2 12

Area Resample Filters

[043] The resulting images may have a slightly blurred appearance, and thus, it may be possible to apply cross-luminosity sharpening filters to substantially correct this:

-8 -8 0	-8 0 -8	0 -8 -8
0 32 0	0 32 0	0 32 0
-8 -8 0	-8 0 -8	0 -8 -8

Cross Luminosity Filters

[044] It will be appreciated that these cross-luminosity filters are distinguishable from cross-color sharpening filters. One possible advantage of cross-luminosity filtering is that blue and white can be sharpened, as well as red and green (as before with cross-color) with a single value, thus reducing the number of operations. In a low cost RGBW implementation, these luminosity values may be calculated using any of the embodiments disclosed in several applications incorporated herein. One example uses the formula:

$$Y = (2 * R + 4 * G + G + B) / 8$$

[045] It should be noted that this luminosity value can be calculated by performing only shifts and adds in hardware or software.

[046] In one embodiment, the color values may be sampled using the area resample filters above, the luminosity “plane” may be sampled using the cross-luminosity filters, and the two results are added together. This can occasionally produce values below zero or above the maximum, so the results may be clamped to the allowed range.

[047] The area resampling filters above correct for the offset position of the sub-pixel within the source pixel with coefficients that sample a little more of the color to one side or the other. An alternative way to accomplish this may be to use a horizontal cubic filter to change the phase of the input data. When an output sub-pixel lands in the center of an input pixel, no phase adjustment is necessary and the centered area resample filter can be used. When the output sub-pixel lands in an offset position in an input pixel, one of the following two cubic filters may be used to generate a “psuedo-sample” that is aligned with the center of the output sub-pixel:

-9 84 199 -18	-18 199 84 -9
---------------	---------------

Horizontal Cubic Filters

[048] Once the phase is aligned, the centered area resample filter and sharpening filter may be used for all output sub-pixels. In one exemplary hardware implementation, these cubic filters may be implemented using special purpose logic to do the multiplies by fixed numbers. This calculation could be done on input values before passing them to the sub-pixel

rendering logic. The sub-pixel rendering logic may thus be simplified, at the cost of the pre-conditioning of the data with the cubic filter. In one exemplary software implementation, it might be advantageous to convolve the cubic filters with the centered area resample filter. This results in two filter kernels shown below:

0	0	4	11	-1	0	0	-1	11	4	0	0
-1	1	77	149	4	-2	-2	4	149	77	1	-1
0	0	4	11	-1	0	0	-1	11	4	0	0

Cubic plus Area Resampling Filters

[049] These two filters can be substituted for the offset filters in the first area resampling case to simulate the cubic case with no other changes to the software. When these filters are used, the luminosity plane may also be phase aligned which might employ convolving the centered sharpening filter with the two horizontal cubic filters:

1	-6	-2	-6	-3	-3	-6	-2	-6	1
0	-2	25	11	-2	-2	11	25	-2	0
1	-6	-2	-6	-3	-3	-6	-2	-6	1

Cubic plus Sharpening Filters

[050] As the layouts of Figures 3A and 3B are similar to the conventional RGB stripe layout, one low cost system might proceed by copying or assigning the nearest RGB or W value into the output sub-pixel without performing area resampling. However, undesirable color error might occur. The horizontal component of this error may be reduced by using the horizontal cubic filters above. As this system would require no line buffers, low hardware costs reduce the overall cost of the system. Additionally, as the cubic filters have a slight sharpening effect, separate sharpening may not be needed. The horizontal lines of fonts may look reasonably good, however the vertical components of fonts may still exhibit color error. Such a low cost system might be acceptable in an image-only application, such as a camera viewfinder.

[051] Figures 5A and 5B are yet other embodiments of a high brightness RGBW layout – but have a 1:2 aspect ratio for their subpixels. This subpixel repeating group comprising blue sub-pixels the same size as the red and green and adding two white subpixels tends to result in a color-balanced RGBW layout. It will be appreciated that the layouts of Figures 3A, 3B, 5A, and 5B – while placing the red and green subpixels and the blue and white subpixels, or red and blue subpixels and the green and white subpixels, on a checkerboard pattern – may be viewed as having other patterns alternatively. For example, any mirror image or rotation or other symmetries are contemplated. Additionally, the subpixels need not be placed on a fully intertwined checkerboard for the purposes of the present invention, an example of which is given in Figure 7.

[052] In one embodiment, each input pixel image data may be mapped to two sub-pixels. In effecting this, there are still a number of different ways to align the input pixels and generate the area resampling filters. The first considered was to simply align 4 input pixels

directly with the layouts shown in Figures 5A and 5B. Figure 6 shows one example of an area resampling of the red color plane as described. Input pixel image data is depicted on grid 602 and the repeating group 604 of subpixels of Figure 5A is superimposed upon the grid. Red subpixels 606 and 610 and their associated “diamond” filters 608 and 612 are also shown. Area resampling may then occur in the manner described herein and in many applications incorporated herein , an example is given here:

$$\begin{array}{ccccccccc}
 -.0625 & 0 & -.0625 & & 0 & .125 & 0 & & -.0625 & .125 & -.0625 \\
 0 & .25 & 0 & + & .125 & .5 & .125 & = & .125 & .75 & .125 \\
 -.0625 & 0 & -.0625 & & 0 & .125 & 0 & & -.0625 & .125 & -.0625
 \end{array}$$

$$\text{DOG Wavelet} \quad + \quad \text{Area Resample} \quad = \quad \text{Luminance Sharpening Kernel}$$

[053] For non-band-limited images, such as text, computer aided drafting (CAD), line art, or other computer generated images, it may be advantageous to treat pairs of subpixels as though they were substantially coincident, using the substantially exact same filter kernel to resample the image. This will result in sharp verticals and horizontal lines being reconstructed.

[054] Alternatively, these diamond filters may be offset by $\frac{1}{4}$ of an input pixel. For a panel with the arrangement of Figure 5A, the filter kernels, shown below, may be substantially the same for red and green; while blue and white use filters may be offset in the opposite direction horizontally.

4 28 0	0 28 4
64 120 8	8 120 64
4 28 0	0 28 4
Red/green	Blue/white

[055] Another embodiment might offset the input pixels until their center points are aligned with the centers of some of the repeating sub-pixels. One example of filters that may suffice are as follows:

0 32 0	16 16
32 128 32	96 96
0 32 0	16 16
Reg/green (or blue/white)	Blue/white (or red/green, respectively)

[056] One of these is the “diamond” filter while the other is split down the middle. This split may results in a blurring of the information in two of the primaries. In one embodiment, by assuming the input pixels are offset $\frac{1}{4}$ pixel to the left, the red and green sub-pixels become perfectly aligned while the white and blue sub-pixels use the split filter. In another embodiment, it may be possible to align the pixels with the highest luminosity, so if the input pixels are assumed to be offset $\frac{1}{4}$ pixel to the right then the white and blue sub-pixels are aligned while the red and green sub-pixels are split across an input pixel. The assignment of the

above filters would be modified for a panel based on the arrangement of Figure 5B, as would be obvious from this teaching to one skilled in the art.

[057] This split may be further processed by using a cubic filter to move the phase of the input data for the split sub-pixels until they are also centered. This may be accomplished by using the following cubic filter to do this $\frac{1}{2}$ pixel offset:

-16 144 144 -16

$\frac{1}{2}$ input pixel cubic offset filter

[058] This offset filter may be easy to implement as shifts and adds in hardware or software. The input pixels are assumed to be shifted $\frac{1}{4}$ pixel one direction for half of the output sub-pixels and they may be rendered with the diamond filter. The other 4 sub pixels may have their input shifted with the above cubic filter then they may also be rendered with the diamond filter.

[059] In hardware, it is easy to implement the above cubic shift on the input data as it flows through the SPR controller. In software, it is often more convenient to convolve the cubic filter with the diamond filter and perform a single filtering operation on the input for the non-aligned sub-pixels. In this case, the following combined filter kernel is used:

0	-2	18	18	-2	0
-2	10	88	88	10	-2
0	-2	18	18	-2	0

[060] For the cases when the sub-pixels are aligned or brought into alignment with cubic filters, the standard cross-color or cross-luminosity sharpening filter may be used. If, however, the input pixels remain centered around pairs of output sub-pixels, then it is possible to use the following cross-luminosity filters for sharpening:

-28 0 -4	-4 0 -28
0 72 0	0 72 0
-28 0 -4	-4 0 -28

[061] Figure 7 is yet another embodiment of the novel high brightness layouts made in accordance with the principles of the present invention. It may be seen that the red and green – as well as the blue and white – subpixels are created on a checkerboard pattern. It will be appreciated that the similar filters as described above may be used on this alternative, although they may be used in a different order or slightly different filter kernels than the other layouts.

[062] Figure 8 is yet another embodiment of a high brightness color filter arrangement as made in accordance with the principles of the present invention. In Figure 8, the subpixels are shown (in grid 802) having its colored subpixels with a 2:3 aspect ratio but white sub-pixels

with an aspect ratio of 1:3. In this embodiment, arranging three rows of three color pixels in a mosaic or diagonal stripe arrangement, the layout becomes color balanced. It should be noted that, with a narrow white subpixel next to each color sub-pixel, each logical pixel has a bright luminosity center. In one embodiment, the input pixels may be centered on these white sub-pixels, so the white value may be simply sampled at each input location. All the color sub-pixels may be split in this alignment, but due to the diagonal stripe layout, the area resampling filter may be a tilted hexagon as in Figure 9.

[063] Looking at Figure 9, input image data grid 900 is shown. Superimposed on grid 900 is target subpixel grid 802. Centers of red subpixels and their associated resample areas (centered around dots 902a, 902b, and 902c) are also shown. In one embodiment, the hexagonal resample areas may be calculated by considering the surrounding red subpixel centers and drawing even boundaries lines between the centers. For example, red center 902a and its associated resample area has a boundary line 906 which substantially bisects the line between center 902 and red center 904. Similarly, lines 908 and 910 substantially bisect the lines between center 902a and 902b and 902c respectively. It will be appreciated that other resample area shapes may be formed in other manners for other embodiments. It suffices that the resample areas are substantially correlated with input image data in a spatial manner. It will also be appreciated that the green color plane – or any other color plane – may be treated similarly.

[064] The resulting filter kernels may be identical for every sub-pixel of every color and could be a 4x3 filter. However, when converted to 8bit integers, the small areas on the

right and left became very small and may be discarded, resulting in the following exemplary filter:

40 12

76 76

12 40

[065] Alternatively, the $\frac{1}{2}$ pixel cubic offset filter may be used to adjust the phase of the input pixels until the psuedo-samples land on the centers of the output sub-pixels again. In this case, the area resample filters may become a 3x3 filters, as given below. Once centered like this, it is possible to use a cross-luminosity sharpening filter for this alignment, as given below.

16 35 0	-16 0 0
35 84 35	0 0 -35
0 35 16	0 102 0
	-35 0 0
	0 0 -16
Area Resampling	Cross-Luminance Sharpening

[066] As with the other layouts disclosed herein, the cubic interpolation accomplishing the $\frac{1}{2}$ pixel alignment may be done on a scan-line basis and may be done to the input data as it arrives. However, in software implementations, it may be convenient to convolve the cubic filter with the above two filters to do each sample in a single step. In this case, the combined cubic and area resampling filter is given below on the left with the combined cubic and sharpening filter on the right:

-1 7 29 19 -2 0	1 -9 -9 1 0 0
-2 14 64 64 14 -2	0 0 2 -20 -20 2
0 -2 19 29 7 -1	0 -6 58 58 -6 0
	2 -20 -20 2 0 0
	0 0 1 -9 -9 1

[067] In another embodiment, the layout of Figure 8 may use a cubic area resampling filter above, but may use a non-cubic cross-luminosity filter. This filter may be desirable for images with sharp edges such as text.

Sub-Pixel Rendering Filters and Offset Assumptions

[068] Apart from use on high brightness layouts, the techniques of performing image data offsets to achieve advantageous filter kernels is also applicable to the full range of other subpixel layouts (e.g. 3-color, 4-color, 5-color, etc.) disclosed herein and in the applications

incorporated by reference. The technique of area resampling may be thought, in one sense, in a geometric model for calculating the filter kernels. A picture of a target layout may be typically drawn on top of a grid of source RGB pixels. A center point, called a resample point, may be chosen for each of the sub-pixels in the target layout. Shapes, called resample areas, may be drawn which enclose substantially all of the area that lies closer to one resample point than any other of the same color.

[069] Figure 10 depicts a three-color subpixel repeating pattern 1000 that substantially comprises red 1002, green 1004 and blue 1006 subpixels of approximately the same size. Grid lines 1008 depict an overlay of source input image data that should be remapped to the target subpixel layout. As may be seen, the input image data grid seems to split the blue subpixels in some ratio (e.g. one half). In the case of the layout of Figure 10, these blue resample areas are simple rectangles. The resample points for red and green were chosen to make the resample areas turn out to be diamonds, or squares rotated 45 degrees as shown in Figure 11. In both squares and diamonds, the shapes of the resample areas were simple enough that the intersection of the areas of the source pixels and the resample areas could be calculated analytically or geometrically.

[070] These choices for red and green resample points are in some sense a simplification, done to make the resample areas easier to calculate and the resulting filters less expensive to implement in hardware. In these filter designs, the resample points of the red and green sub pixels were not placed at the centers 1102 of the sub-pixels, but were moved slightly left or right to make them align with the centers 1104 of the source pixels or logical pixels, as seen in Figure 11. If these resample points are placed substantially at the centers 1102 of each

target sub-pixels, then the resample areas become more complicated asymmetrical diamond-like shapes, as seen Figure 12. These shapes sometimes resemble kites flying sideways -- so the resulting filters are termed "kite filters". These new shapes may be more difficult to calculate geometrically and they may change with every variation of any given subpixel layout. In some cases, it may be advantageous to leave the resample points substantially on the center of the subpixels. For example, this may reduce color error in some images. In other cases, it may be advantageous to move the resample points substantially to the center of the resample area. For example, this may simplify the filters and make implementing them in hardware less expensive.

New Filter Generation:

[071] One embodiment of generating resample areas and their filter kernels will now be described:

- (1) A first step is to accept a list of resample points and create a picture or other representation in a bitmap file.
- (2) Each pixel in this image is substantially compared against all the resample points to find out which resample point is closest. In doing this, it may be desirable to consider all neighboring resample points above, below, left, right as well as in all four diagonal direction.
- (3) A second pass through the bitmap image may be taken and the count of the number of pixels that are tagged as closest to one resample point may be an approximation of the resample area for that resample point. The number of tagged pixels inside each source pixel square may also be counted.

- (4) The ratio of these two numbers may be an approximation of the coefficient for the filter kernel for each source pixel. The bitmap image can be displayed or printed out to see what the resulting shape looks like and to verify that the shapes make sense.

[072] It will be appreciated that other methods and steps may be taken to generate filter kernels for the mapping of input image data to a target subpixel layouts. It suffices for the purposes of the present invention that the filter kernels extract out image data that is substantially correlated to the target subpixels in a spatial manner.

Translating Edge Assumptions:

[073] Using a point near the exact center of the sub-pixel as the resample point, in some cases, may be simplified by changing the edge assumptions. A simplifying assumption of placing a target layout (such as shown in the '353 application and other applications incorporated herein) on top of 4 source pixels may result in diamonds and boxes that may be out-of-phase with the input pixels. One example is seen in Figures 13A, 13B, and 13C – depicting the red, green and blue resample areas respectively. Translating all the resample points together is not a simplification since the choice of edge alignment could be arbitrary. In many of the layouts, a slight shift to the left of all the resample points resulted in much simpler filters and sharper greens. For example, such suitable shifts result in the resampling areas seen in Figures 14A, 14B, and 14C.

Adjusting Center Locations:

[074] When points close to the exact sub-pixel center are used as the resample points for the layout shown in Figure 15 (e.g. two blue subpixels 110 staggered within a substantially checkerboarded pattern of red and green subpixels), a large set of different filters may result. For example, Figures 16A, 16B and 16C are one possible set of filters for such a layout.

[075] In another embodiment, both the red resample points can be moved slightly to make the red filter areas diamonds, as may be seen in Figure 17A – with Figures 17B and 17C depicting the green and blue filters respectively. Yet another embodiment might be a combination of translation and adjustments to make the two green areas the diamonds -- while the red and blue would remain “kites”, as is shown in Figures 18A, 18B and 18C. This may have the effect of keeping green sharper. Since green has most of the luminosity, this may result in a sharper total image. In addition, having all the green resample points centered on input pixels would allow them to be sharpened with cross-color sharpening.

Decimation Filters:

[076] Adjusting the relationship between source pixels and the subpixels in the layout shown in Figure 15 might also help with decimating RGB data into such a display. As may be seen in Figure 15, there may be a red or a green sub-pixel completely inside each source pixel. In a simple-to implement hardware decimation mode, the correct red or green primary value from the underlying RGB pixel could be copied directly into the target sub pixels. The blue sub-pixels may be split and may be averaged or even have one of the two source blue values used arbitrarily without noticeable problems in the image.

[077] If the edges of the source pixels are aligned with the target layout, one of the green sub pixels may be split between two source pixels. Averaging the two source greens may produce a fuzzy image; while picking one source value may result in some degradation of image quality. Alternatively, the remapping grid 1502 could be shifted between the source pixels so that the green sub pixels are not split, as may be seen in Figure 15. This will result in one of the red sub pixels being split between two source pixels, but since green contributes more to the luminosity of the image, splitting one of the reds may not degrade the image as much.

[078] While the invention has been described with reference to an exemplary embodiment, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted for elements thereof without departing from the scope of the invention. In addition, many modifications may be made to adapt a particular situation or material to the teachings without departing from the essential scope thereof. Therefore, it is intended that the invention not be limited to the particular embodiment disclosed as the best mode contemplated for carrying out this invention, but that the invention will include all embodiments falling within the scope of the appended claims.